Desafio Técnico Back-End

Introdução

Esse é um teste técnico para a vaga de Back-End, com isso, estamos aplicando um desafio que fará parte da sua rotina, caso integre o nosso time.

Como Desenvolvedor Back-End (Python/Django), você fará parte da equipe de desenvolvimento de software da Weni by VTEX, uma unidade de negócios da VTEX. Sua responsabilidade será desenvolver aplicações, funcionalidades ou corrigir bugs em aplicações mantidas pela empresa, garantindo seu pleno funcionamento. Você definirá a arquitetura e as funcionalidades das aplicações, ajudando-nos a tomar decisões técnicas importantes e a avaliar seu impacto no produto a médio e longo prazo.

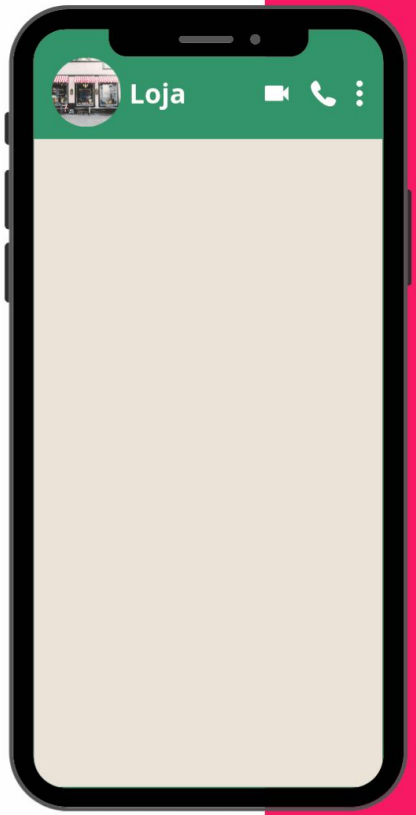
Aviso: É **expressamente proibido compartilhar esse documento** com outras pessoas sem a expressa autorização da Weni by VTEX.

Teste Técnico

Sobre a Weni

A Weni by VTEX é uma empresa de Inteligência Artificial aplicada à comunicação, focada em uma solução enterprise de customer experience baseada em IA para marcas e varejistas.

Com a Weni Plataforma, ajudamos a redefinir a jornada de venda e de pós-venda através de hiper automação e dados integrados aos canais conversacionais.



Agent Builder

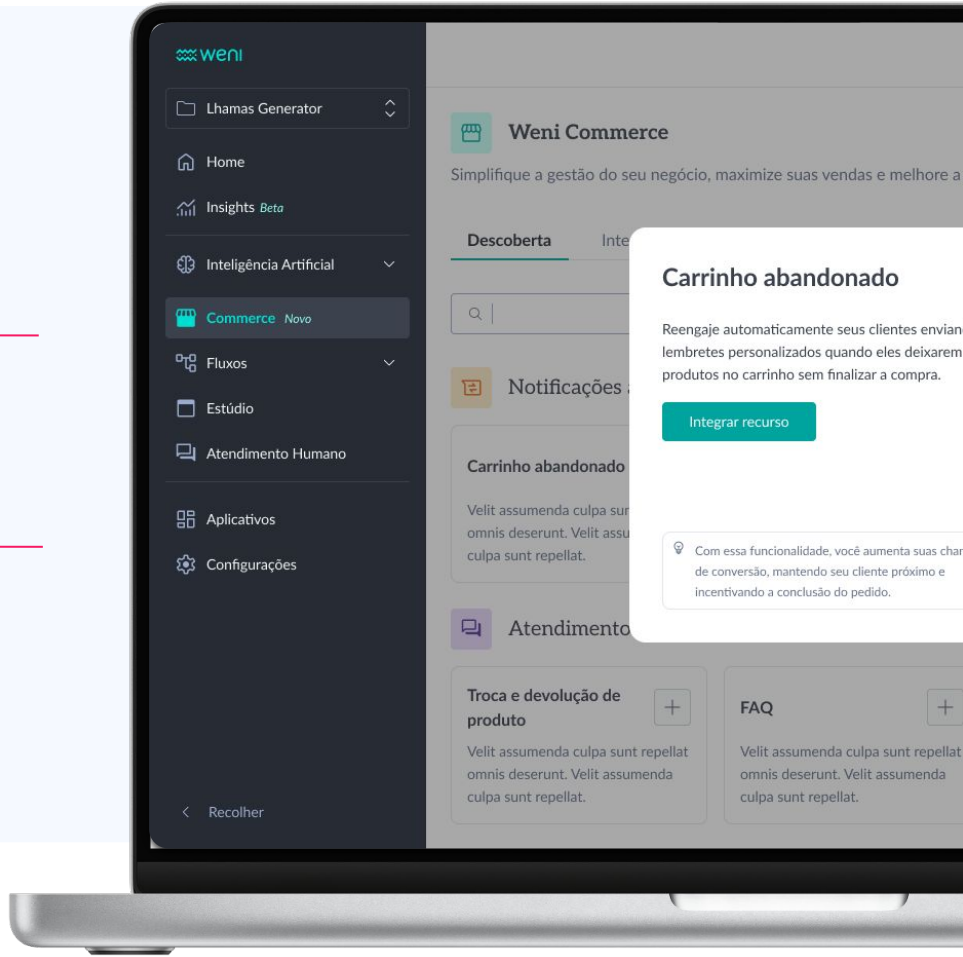
Módulo da Weni Plataforma que permite criar agentes inteligentes, personalizados e conectados à base de conhecimento da empresa.

Activation

O módulo de Commerce da Weni simplifica e agiliza a gestão de processos de comércio digital. Com apenas alguns cliques, os clientes podem ativar soluções que notificam automaticamente o consumidor final em diferentes etapas do funil de compra.

Weni Chats

Módulo de atendimento humano da Weni Plataforma que permite que os cliente Weni crie, personalize, atenda e administre contatos e conversas.



Applications

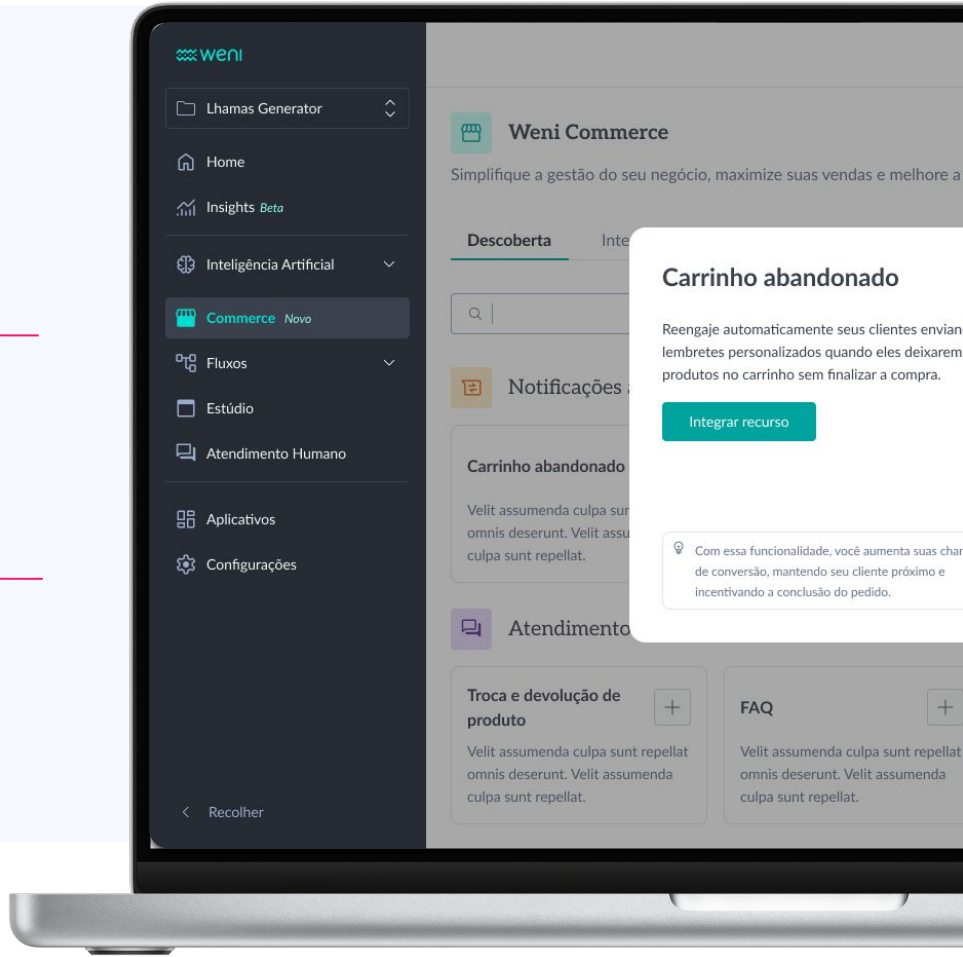
Módulo da Weni Plataforma em que é possível realizar integração com diferentes ferramentas de comunicação e gestão do mercado.

Flows e Studio

O Flows é o módulo da Weni Plataforma que permite criar fluxos conversacionais personalizados do zero, controlado as ações do seu agente. Já o Studio oferece recursos para organizar contatos em grupos e criar fluxos segmentados por meio da funcionalidade de campanhas.

Insights

Módulo da Weni Plataforma que fornece dados e métricas essenciais sobre os atendimentos em andamento e encerrados.



weni by VTEX

Teste Técnico

Contexto

O propósito desse teste é criar uma aplicação web com principal foco no back end, utilizando a linguagem Python e o framework web Django. Para tanto, os requisitos representam um cenário hipotético que pode ou não ser usado na vida real.

Problema:
Muitas empresas enfrentam desafios significativos ao tentar atender eficientemente seus clientes por meio das diversas plataformas de comunicação disponíveis. A multiplicidade de canais, como Telegram, Facebook, WhatsApp e etc., introduz uma complexidade adicional na gestão das interações.

Solução:
Para resolver o problema dessas empresas, você precisa desenvolver um sistema de integração de canais de comunicação, que inclui suporte para os principais aplicativos de mensagens, como Telegram, Facebook, WhatsApp, etc. Esse sistema permitirá que as empresas gerenciam eficientemente as interações com seus clientes, oferecendo respostas automáticas através de um chatbot.

Teste Técnico

Contexto

Requisitos:
Queremos que essa ferramenta tenha uma excelente performance, funcione corretamente e tenha um código claro. segue abaixo os principais requisitos:

Sua ferramenta precisa ser uma ponte entre os Atendentes Humanos e os seus clientes(Contatos) a partir dos canais de comunicação.

Fluxo 1 - Contato para atendente humano

1. Um contato envia uma mensagem para um chatbot através de um dos canais de comunicação configurados;
2. O canal de comunicação, vamos tomar o Telegram como exemplo, envia uma chamada webhook contendo o conteúdo da mensagem;
3. O sistema processa a mensagem e a persiste em seu banco de dados

Teste Técnico

Contexto

Fluxo 2 - Atendente humano para contato

1. A partir de uma API REST fornecida pelo seu sistema um atendente humano envia uma requisição contendo um identificador do contato e mensagem;
2. O sistema persiste a mensagem e a envia para o contato pelo respectivo canal.

Requisitos

- Camada de abstração: Crie uma camada de abstração para a integração de canais de comunicação, de modo que seja possível trocar ou adicionar um canal facilmente sem afetar o restante do sistema.
- Canais: Integre pelo menos um canal de comunicação real e um mock (implementar mais canais será um grande diferencial). Recomendamos o Telegram devido à sua popularidade e facilidade de integração. Como criar um chatbot do Telegram com Python.
- Persistência: Persistir pelo menos *Contato*, *Canal*, *Atendente Humano* e *Mensagem.*

Teste Técnico

Contexto

- APIs REST:
  Desenvolva as seguintes APIs REST utilizando o Django Rest Framework:
  - API que permita integrar um canal com base em seus critérios de configuração;
  - API que permita enviar uma mensagem para um contato específico a partir de um identificador;
  - Uma API Webhook para cada canal permitindo assim que eles enviem mensagens para o seu serviço;
  - API para listar as mensagens. É muito provável que com o tempo sejam criados milhares de mensagens, portanto, essa deve ser paginada. É importante também que seja possível filtrar as mensagens por um *Contato* ou por um *Atendente Humano* específico.

- Testes unitários: Cubra sua aplicação com testes unitários, pois eles são como o alicerce de um edifício. Um software sem eles é como uma estrutura sem base sólida - suscetível a colapsos e dificuldades de manutenção.

Teste Técnico

Contexto

Lembre-se: Software que só funciona na máquina do desenvolvedor é um produto inacabado. Como todo projeto, você precisa se preocupar com a manutenção, que pode ser feita por você ou por outros desenvolvedores, e uma parte importante para uma boa manutenção é uma documentação simples, clara e objetiva (um nome claro para uma classe, método ou função dispensa comentários). O formato e a organização da documentação ficam por sua conta.

Diferenciais:

Esses recursos não são obrigatórios, mas, ao implementá-los você se destaca dos demais candidatos.

- Autenticação: Crie um mecanismo de autenticação utilizado o padrão JWT. Autenticar as APIs REST é muito importante para assegurar que os usuários que acessam o sistema de fato possuem a devida permissão;
- Cache: Você pode reduzir significativamente o esforço dos Atendentes Humanos salvando perguntas frequentes no cache;

Teste Técnico

Contexto

- Poetry: Utilizamos Poetry no nosso dia a dia, utiliza-lo como gerenciador de pacotes será um diferencial;
- Docker: Implemente um Dockerfile e um docker-compose para facilitar a execução de sua aplicação;
- Celery: Crie uma rotina que apaga mensagens criadas a mais de um mês, evitando assim um grande volume de dados;
- Atendimento Humano Funcional: Integre uma plataforma real de atendimento humano ao seu projeto, faça com que seja possível trocar mensagens entre seus canais e essa plataforma;
- Deploy: Faça deploy da sua aplicação em alguma plataforma como o PythonAnywhere, por exemplo.

Fique à vontade para implementar recursos inovadores além dos citados acima, pois isso irá enriquecer ainda mais a experiência do usuário e fortalecer a posição do seu projeto como uma solução diferenciada dos demais candidatos.

Teste Técnico

Contexto

Recomendações:

- Leia o zen do python (import this);
- Utilize git-flow;
- Configure seu editor para usar um lint que siga o PEP8;
- Seu primeiro commit deve ser o arquivo . gitignore;
- Desenvolva um Dockerfile para facilitar o deploy;
- Siga os princípios do S.O.L.I.D.
- Leia nosso blog sobre Como funcionam os bots no Telegram? .

Teste Técnico

Entrega

Para a realização do desafio, você terá um prazo que começa a contar a partir do dia que você recebeu esse documento no seu email. Caso você tenha algum tipo de impedimento, entre em contato conosco e descreva seu problema. Garanta que você entendeu claramente todos os requisitos. Foque em entregar um projeto com a qualidade que você considere adequada.

Nos envie o teste mesmo que não conclua todos os requisitos.

Seu prazo final para entrega será em 21/08/2025 (quinta-feira) até às 23:59h.

Entrega:

O código deverá ser colocado em um repositório privado no Github ou Bitbucket. Lembre-se de dar acesso de leitura para o usuário: sandro.meireles@vtex.com.

Ao finalizar o teste envie um email para: mariana.cruz@vtex.com informando que o teste foi concluído.

Dúvidas?

Mariana Cruz
People Partner Weni by VTEX
mariana.cruz@vtex.com

weni
by VTEX